Ministry of Information Technologies and Communication named Muhammad Al-Khwarizmi Information Technologies University

Faculty: Computer engineering
Department: Information technology
"Web technology"

Course work

Theme: **Creating online a task system for teachers and students**

Group: 222-16
Rare fulfilled: Toxirov Behruz
Leader: _____
Protection: _____ (date)
_____(Ratings)
_____(Signature)
_____(Signature)

2020

Ministry of Information Technologies and Communications Ministry for Development
Muhammad Al-Khwarizmi Tashkent University of Information Technologies

Faculty: _____
Department: _____
Direction (Specialist): _____

I approve
The department _____
>> << _____ _____ 2020

Course work
Assignment.

1) Course on
2) Rare 222-16 group Toxirov Behruz.
   Chair: _____
3) Creating online a task system for teachers and students
   Topic No. 1 Protocol approved by the department meeting.
3) The first information on the launch.

Your job is to chart:

№	**Designations**	Beginning	the end
1.	The theoretical part		
2.	Applied part		
3.	The main part		
4.	conclusion		

4) List of issues to be considered the content of the explanatory note to calculate the _____
5) List of graphic material: _____

Paper: _____
Leader: _____

Theme: **Creating online a task system for teachers and students**

CONTENTS

INTRODUCTION

The Decree of the President of the Republic of Uzbekistan “On measures to further improve the project management system in the field of information and communication technologies” No. PP-3245 dated 29.08.2017 speaks about expanding the market of information services, introducing information systems and networks of inter-agency electronic interaction and information exchange between government agencies and other organizations, including in the framework of the Unified National Information System of Project Management of the Republic of Uzbekistan. This allows various organizations and structures to more widely implement information technologies in various spheres of human activity. Currently, many universities are creating systems for winning. for example, an online rating system. Active usage of information technologies in every field of human activity nowadays is not a matter of choice. So any higher education institution as well as any other organization is forced to use information systems in order to be in the flow. So, it should be noted that real pace of changes in global education sector (especially its private part) is much quicker than stereotypes. Thus, higher education institutions eagerly try to meet these changes, though in most cases higher education institutions’ infrastructure and information systems fail to do that. Consequently, there is a pressing scientific task of redefining infocommunication needs of higher education institutions through their information flows in order to identify actual modern requirements to the higher education institutions’ information systems and infocommunication infrastructure. Those specific needs and requirements are identified on the basis of analysis of cutting-edge information technologies and trends that are used or expected to be used in education. As the result specific recommendations on infocommunication flows and university information systems integration are proposed. While the reference to homework might make you think this is more for school students, it’s actually a perfect app for all levels, from high school to university.

CHAPTER I. THEORITICAL PART. ABOUT TASKS SYSTEM INFORMATIONS

1.1. TASKS SYSTEM INFORMATION

Evernote and OneNote are apps designed specifically for note taking. If you're a long-time Microsoft Office user, you should find yourself comfortable in OneNote. But for Google Docs advocates like myself, Evernote might feel more familiar—it's the app I'm in for at least 10 hours each week, so I'll focus on it here. Evernote lets you create "notes," which could be text, pictures, voice messages, videos, PDFs, and almost anything else you can save. You can sort those notes into notebooks, create stacks of notebooks that fit a certain theme, and tag notes to organize them into categories. You can even create to-do lists in Evernote to help you, for example, stay on top of reviewing notes for upcoming test dates. The best feature for amplifying your note taking, though, is Evernote's search capabilities. Just type in whatever you're looking for in the search box, and Evernote shows every related note on that topic—you'll never again have a hard time finding the information for that concept you forgot. OneNote is also ideal for students. If you have a tablet PC, you can sketch or write by hand directly on your device—and OneNote will let you convert that to text or even make your handwriting searchable. Snap photos of whiteboards or slides to quickly capture that information for later. Record audio directly in OneNote and you can also search those audio files (nifty!). Perhaps OneNote's most compelling feature is its interface: You can organize it just like you would a binder, with notebooks you can create for each class and tabs for things like assignments or lecture notes. Or just use one student notebook and use the tabs to organize each subject. Teachers can even create Class Notebooks for their students to organize course content.

Evernote and OneNote Tips

Create a notebook for each class, and then keep all of your class notebooks in one "Classes" notebook stack. Within each notebook, create a new note for each lecture, assignment, and reading.

- If you're working with a group, consider creating a "shared" notebook so that you all have your notes in one place.
    - Connect your Evernote or OneNote account to app integration tool Zapier to, for example, automatically add reminders to your calendar. You can even copy notes between these two apps. The two-year project started in 2016 and was funded through a Strategic Priority Commissioned Grant by the Office of Learning and Teaching (OLT) of the Australian Government, and addresses the challenges faced by Australian Higher institutions to improve the quality of the student experience within the context of increasing enrolment numbers over the past years. The project is led by The University of Sydney (USYD) in collaboration with University of Technology Sydney (UTS), University of New South Wales (UNSW Australia), University of South Australia (UniSA), University of Texas at Arlington, and The University of Edinburgh.
    - As part of its deliverables, the On Task project developed a software tool that gathers and assesses data about student's activities throughout the semester and allows instructors to design personalised feedback with suggestions about their learning strategies. By providing frequent suggestions about specific tasks in the course, students are able to quickly adjust their learning progressively.
    - The tool is LMS agnostic and receives its data from various sources such as video engagement, assessments, student information systems, electronic textbooks, discussion forums, etc. Instructors and educational designers can use the platform to connect large data sets about students with concrete and frequent actions to support their learning.
    - Examples of feedback On Task could facilitate includes directing them to specific chapters or worked examples in their textbook, suggesting additional reading or resources, enrolling them in required workshops or laboratory tutorials, suggesting the most effective study techniques for the tasks in the course, directing them to university support services, etc. On Task aims to assist instructors to support all students in a course regardless of their performance by providing relevant, personalised suggestions.

- The tool is also designed to provide evidence to management bodies about student support actions and their impact on the overall learning experience. Its open and modular architecture is conceived to foster a shift at the institutional level on how to improve the quality of the learning experience.

1.2. Example of tasks system

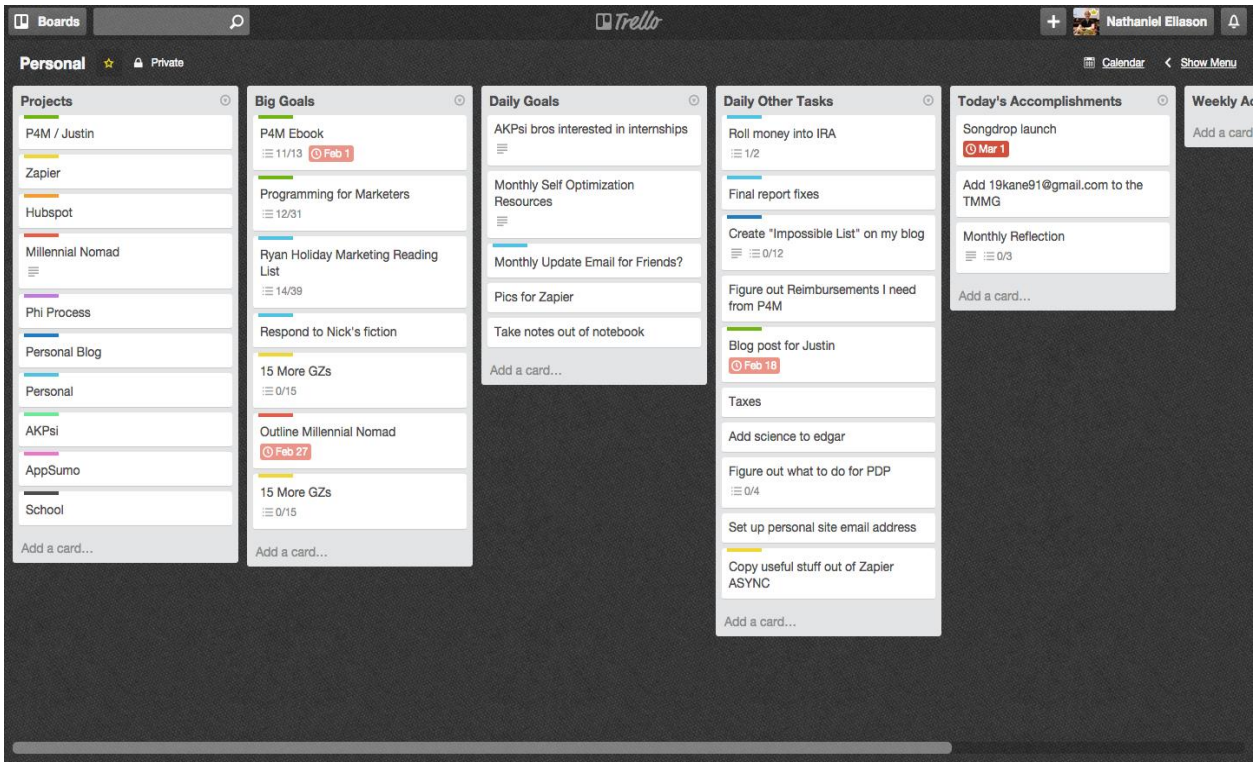
A great app for keeping your thoughts as well as your assignments organized, Evernote allows you to sync personal checklists and notes across devices, meaning you can work on a task on one device and later switch to another without losing anything. You can take notes in a variety of formats, including text, photos, audio, web clippings and videos, and can attach Microsoft Office documents and PDFs. It also lets you work collaboratively with others and share ideas, as well as plan events and set reminders If you need to revise or get started on an essay, you might want to try the Pomodoro Technique, in which you can break up your work into intervals of 25 minutes, taking a short break after each one. This technique is proven to make you less likely to burnout. You only need a timer to do this, but there are apps available to time the 25 minutes and let you track your productivity, such as Pomodairo, an Adobe Air app that lets you mark where you're getting distracted and see how long you've spent on different tasks. Or, if you'd prefer a desktop version, you could download Tomighty. If you're hooked on social media and find yourself checking Twitter five minutes into a revision session, you may want to download an app to block yourself from going on social media or any other distracting websites. For Google Chrome users, Stay Focusd is a highly rated extension which lets you restrict the amount of time you can go on time-wasting websites. Or, for your phone, you could use Anti-Social, which lets you see how much you use your phone compared to others, and allows you to block apps you overuse. There are also many time management apps for students that let you check off your virtual to-do list, ideal if you ever feel overwhelmed with tasks. One such app is Remember the Milk, which can sync with all your devices and integrate with your calendar, emails, Twitter and other time management tools. Or, if you're a

visual person, you might prefer 2Do, which uses color coding and lets you categorize tasks by priority and subject.

Another ideal to-do list app for procrastinators is Finish, which gives you a nice rewarding sound and checkmark feature when you complete tasks.

One of the most unique apps on this list is Coach.me, in which you'll join a community of people working to achieve individual goals. You'll get support from others to help you form good habits and make yourself more productive, earning 'props' (similar to Facebook likes) from other users in recognition of your achievements. The app has helped over a million people form new habits and has earned plenty of praise, with one reviewer saying: "This app has been a wonderful resource to track individual growth and development. Thank you for making such a great product, it has helped me become a better person and tracked my progress towards my goals."Available for both iOS and Android, Google Keep is a note-keeping app that uses a pin board format, allowing you to pin notes, lists, photos and voice memos, easily search for previous pins, share your list with others and receive location-based reminders. It's well known that flashcards can be an excellent tool for helping to memorize important information for exams, and Quizlet lets you create your own flashcards or study using existing ones made by other students. It also gives you hints on where you might need to improve, and has a game called Match in which you race against the clock. You can choose from over 220,000 study sets, and its features are available in 18 different languages. It's well known that flashcards can be an excellent tool for helping to memorize important information for exams, and Quizlet lets you create your own flashcards or study using existing ones made by other students. It also gives you hints on where you might need to improve, and has a game called Match in which you race against the clock. You can choose from over 220,000 study sets, and its features are available in 18 different languages. Last but not least in our list of useful apps for students, Simple Mind + lets you organize your thoughts by creating mind maps, which (using the pro version) can be seamlessly synchronized to your Google Drive or Dropbox and shared with others. You can also enhance your mind maps with photos, videos and voice memos, and easily customize

the appearance and select different layouts to suit you.



Trello is a free project management app that lets you create "cards" for things that need to get done and organize those cards into different lists. If you're familiar with the Kanban project management system, Trello offers a similar solution: you have a list of "To-Dos," a list of things you're "Doing," and a list of things that are "Done." As you make progress, you move cards from one column to the next so you can see your improvement over time. It's a powerful tool for managing long-term projects, since the cards and lists display lots of information at a glance. You can even attach files to Trello cards, assign cards to teammates, give cards due dates, and see a card's progress.

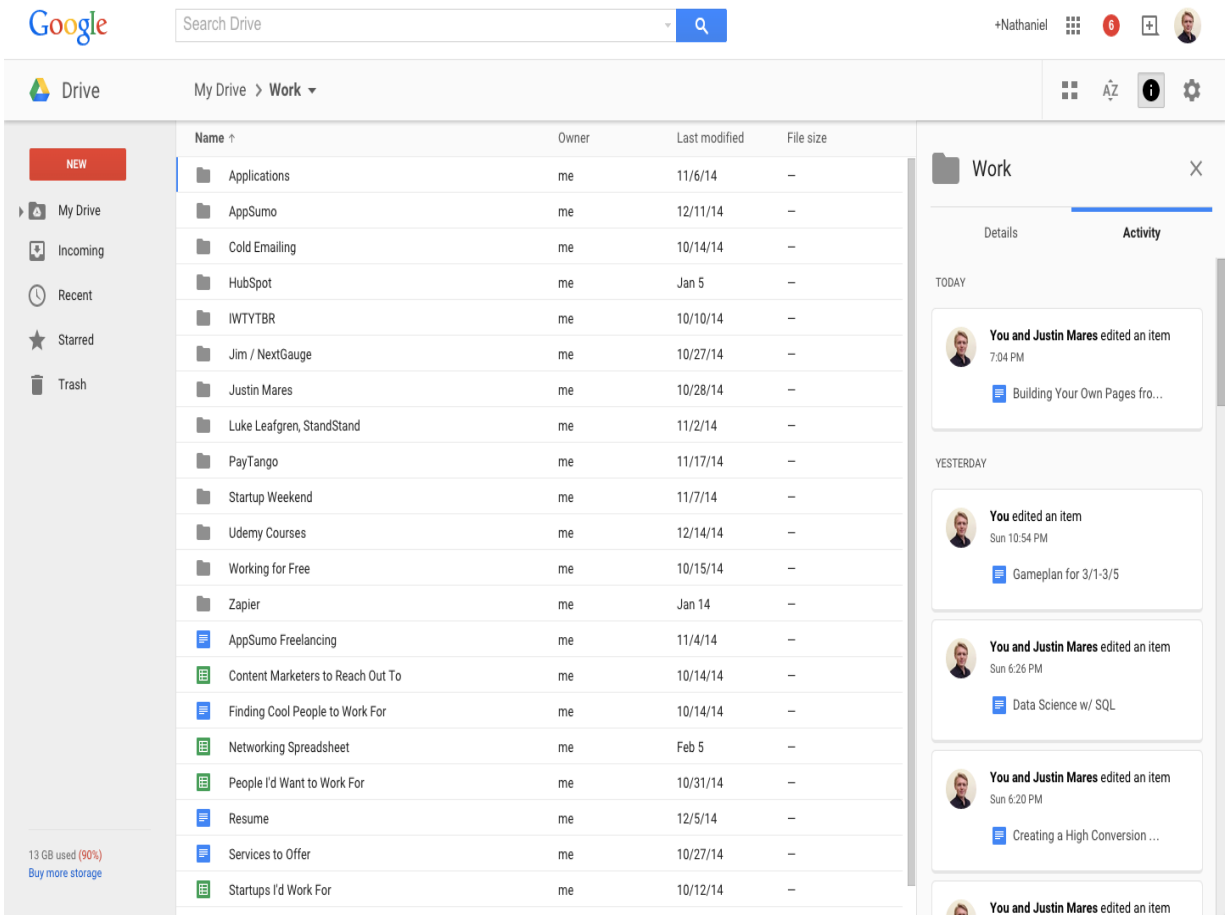
Trello Tips

- When you start a new group project, create a board in Trello specifically for that project and add your team to it.
- Once everyone is on the board, add lists for the different stages, such as "to-do," "doing" and "done" and then start placing cards.

- Use Zapier with Trello to set up recurring cards.

Augmented Reality Scenario Model (ARSM), an augmented reality scene is represented as a graph composed of nodes representing AR-Objects. AR-Objects permit one to describe in a uniform way three categories of entities that can be found in an AR environment, namely, real objects, virtual objects, and scenes comprised of real and virtual objects. The solution enables describing the whole spectrum of augmented reality scenes ranging from real scenes composed of real objects, through mixed scenes composed of both real and virtual objects, to virtual scenes composed merely of virtual objects.

1.3 Advantages and disadvantages of the system



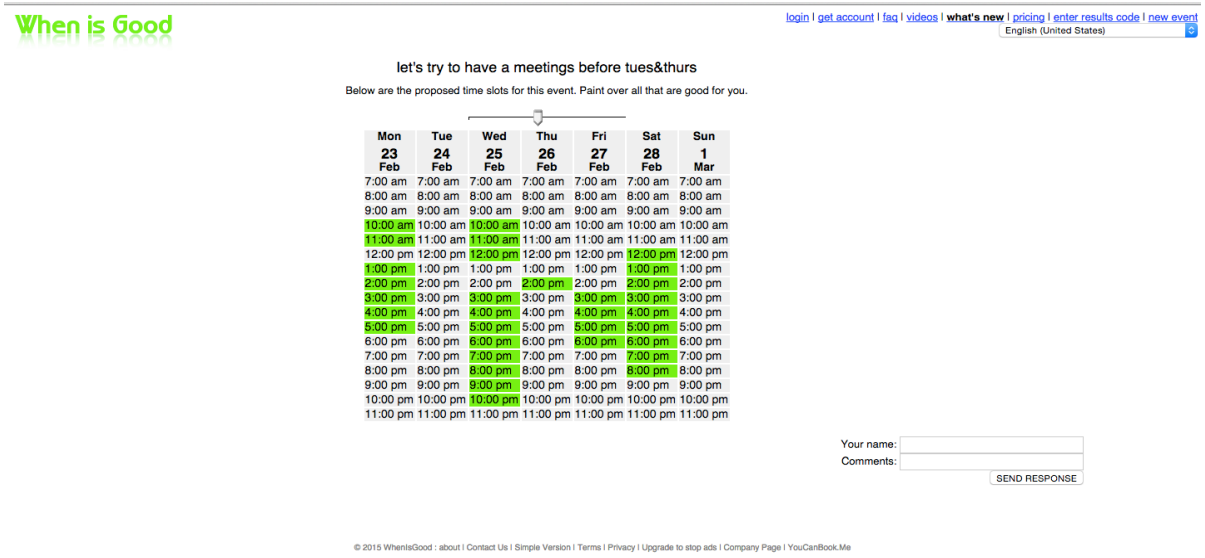
Like Dropbox, Google Drive gives you a folder on your computer that you can store files in that will be automatically backed up online for safe keeping. You can share these documents with others, and create shared folders that collaborators can access. Google gives you 15 GB of free storage to use with Google Drive, Gmail, and Google Photos. The big benefit that Google Drive has over Dropbox is that it's integrated

with Google Docs. That means you can create, share, and edit documents online, and save them directly to your Google Drive folders, instead of just sending documents back and forth.

Google Drive Tips

- Create a shared Drive folder for everyone that you're working with.
- Upload materials that the team needs directly to the shared folder, so everyone can collaborate on them.
- Learn advanced Google Drive search tips to find any file fast.

Doodle and WhenIsGood for Scheduling Meetings



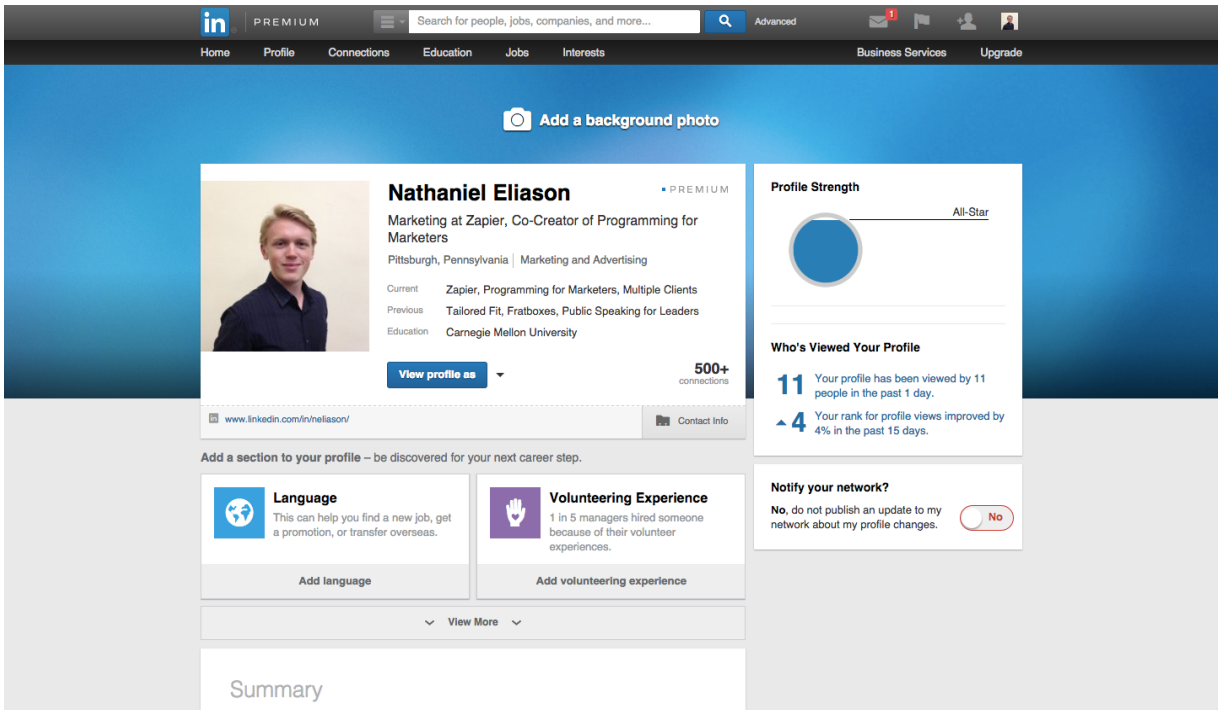
Finding a time that works for everyone is simple with WhenIsGood

Last, you need to figure out when everyone in your group can meet. Doodle and WhenIsGood are by far my favorite tools for determining this. Just set some times that could work for the meeting, and then send the forms to everyone on your team so they can put in their availability.

In Doodle they just check off the times that work, and in WhenIsGood they can click and drag over all their available time slots. Either way, you get fast feedback on the best time to meet.

Doodle and WhenIsGood Tips

- For Doodle, sign up for an account so that the decided upon meeting time is automatically added to your calendar.
- Be generous with your times. Include some times that you can't meet just in case it won't be possible for the whole team to meet.
- Give people lead time to respond to these polls! Don't expect instant responses, so send them out as early as possible. LinkedIn for Finding Every Job Opportunity



Even if you haven't started your career yet, it's a good idea to join LinkedIn

If you're a college student and you're not on LinkedIn, stop reading right now—go sign up! LinkedIn is a great tool for networking with professionals at companies you might be interested in working for post-graduation. Most large companies have HR personnel who regularly check LinkedIn for job candidates. Moreover, you can bet that if you apply for a job or internship, that company will look at your LinkedIn to see what you've done. You can also use it to get introduced to people who might be able to give you interviews or referrals to people looking for interns.

CHAPTER II. PRACTICAL PART. DEVELOPING TASKS SYSTEM

2.1. Database and Datalogic model

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database design involves classifying data and identifying interrelationships. This theoretical representation of the data is called an ontology. The ontology is the theory behind the database's design. In a majority of cases, a person who is doing the design of a database is a person with expertise in the area of database design, rather than expertise in the domain from which the data to be stored is drawn e.g. financial information, biological information etc. Therefore, the data to be stored in the database must be determined in cooperation with a person who does have expertise in that domain, and who is aware of what data must be stored within the system. This process is one which is generally considered part of requirements analysis, and requires skill on the part of the database designer to elicit the needed information from those with the domain knowledge. This is because those with the necessary domain knowledge frequently cannot express clearly what their system requirements for the database are as they are unaccustomed to thinking in terms of the discrete data elements which must be stored. Data to be stored can be determined by Requirement Specification. Once a database designer is aware of the data which is to be stored within the database, they must then determine where dependency is within the data. Sometimes when data is changed you can be changing other data that is not visible. For example, in a list of names and addresses, assuming a situation where multiple people can have the same address, but one person cannot have more than one address, the address is dependent upon the name. When provided a name and the list the address can be uniquely determined; however, the inverse does not hold - when given an address and the list, a name cannot be uniquely determined because multiple people can reside at an address. Because an address is determined by a name, an address is considered dependent on a name. (NOTE: A common 11 misconception is that the relational model is so called because of the stating of relationships between data elements therein. This is not true. The

relational model is so named because it is based upon the mathematical structures known as relations.) Once the relationships and dependencies amongst the various pieces of information have been determined, it is possible to arrange the data into a logical structure which can then be mapped into the storage objects supported by the database management system. In the case of relational databases the storage objects are tables which store data in rows and columns. In an Object database the storage objects correspond directly to the objects used by the Object-oriented programming language used to write the applications that will manage and access the data. The relationships may be defined as attributes of the object classes involved or as methods that operate on the object classes. The way this mapping is generally performed is such that each set of related data which depends upon a single object, whether real or abstract, is placed in a table. Relationships between these dependent objects is then stored as links between the various objects. Each table may represent an implementation of either a logical object or a relationship joining one or more instances of one or more logical objects. Relationships between tables may then be stored as links connecting child tables with parents. Since complex logical relationships are themselves tables they will probably have links to more than one parent. Database designs also include ER (entity-relationship model) diagrams. An ER diagram is a diagram that helps to design databases in an efficient way. Attributes in ER diagrams are usually modeled as an oval with the name of the attribute, linked to the entity or relationship that contains the attribute. Before creating a registration on the internet, an infographic model is created. The purpose of this model is to create a place in the database of each table. The following are the elements (forms) that will be included in the structure of the schema of the Internet store database, where each table is linked and linked: 12 2.1.1table is attitudes Adatabase based on this scheme will be built. Each sequence in this scheme should be shaped in the creation of a database. Relationships are crucial in establishing databases for the rela - tion model. In fact, the database connects objects to their relationships. Basically the database contains the following relationships: 1. Together - one (1: 1) attitude. A set of A and B objects is set to 1: 1, if each copy of A is equivalent to one copy of V object, and

vice versa each copy of object V A if one copy of the app corresponds. 2. Together - many (1: n) attitudes. In A and B objects, each copy of Object A corresponds to more than one copy of object V, whereas each copy of object V fits more than one copy of object A This is the case. 3. If multiple - one (n: 1) relationship is between a set of A and B objects, each copy of Object A is the maximum one copy of the V object. Among the V objects, there are several copies of the A object. 4. Multiple - Many (m: n) Attitudes. The relationship between A and B objects is set if each copy of Object A is multiple copies of the V object, and vice versa

2.2. Workplace in Yii2 framework

The story of Yii began on January 1, 2008, as a project to fix some flaws in the PRADO framework (PHP Rapid Application Development Object-oriented), which became the winner of the Zend PHP 5 coding contest in 2004. The PRADO framework was an attempt to port ASP.NET to the PHP platform, including ViewState, PostBacks, Page_Load and OnClick, which resulted in sections of code simply copied from ASP.NET. For example, the separation of Rare Fields and Occasional Fields in the Control class for the purpose of memory optimization, which makes sense in .NET but is of dubious value in PHP. PRADO inherited from ASP.NET almost all the negative aspects: it slowly processed complex pages, had a steep learning curve, and was rather difficult to set up.

Yii has a compilation of official documentation, such as a tutorial on developing a simple blog application, a guide that describes each function and a link to a class that gives every detail about properties, methods, and events. There is also documentation provided by users, most of which is available on the official website's wiki.

Yii is released under the new BSD license (3-clause license). This means that you can use it for free to develop both open and proprietary web applications. The text contained in the Yii White Paper is licensed to the public under the GNU Free Documentation License (GFDL). In general, the contents of Yii documentation can be copied, modified, and redistributed as long as the new version provides the same freedom to others and recognizes the authors of the Yii documentation used. The logo is licensed under a Creative Commons Attribution-No Derivative Works 3.0 Unported

Licens

Yii is a high performance, component-based PHP framework for rapidly developing modern Web applications. The name Yii (pronounced Yee or [ji:]) means "simple and evolutionary" in Chinese. It can also be thought of as an acronym for **Yes It Is**! Yii is a generic Web programming framework, meaning that it can be used for developing all kinds of Web applications using PHP. Because of its component-based architecture and sophisticated caching support, it is especially suitable for developing large-scale applications such as portals, forums, content management systems (CMS), e-commerce projects, RESTful Web services, and so on.

If you're already familiar with another framework, you may appreciate knowing how Yii compares:

Like most PHP frameworks, Yii implements the MVC (Model-View-Controller) architectural pattern and promotes code organization based on that pattern. Yii takes the philosophy that code should be written in a simple yet elegant way. Yii will never try to over-design things mainly for the purpose of strictly following some design pattern.

Yii is a full-stack framework providing many proven and ready-to-use features: query builders and ActiveRecord for both relational and NoSQL databases; RESTful API development support; multi-tier caching support; and more.

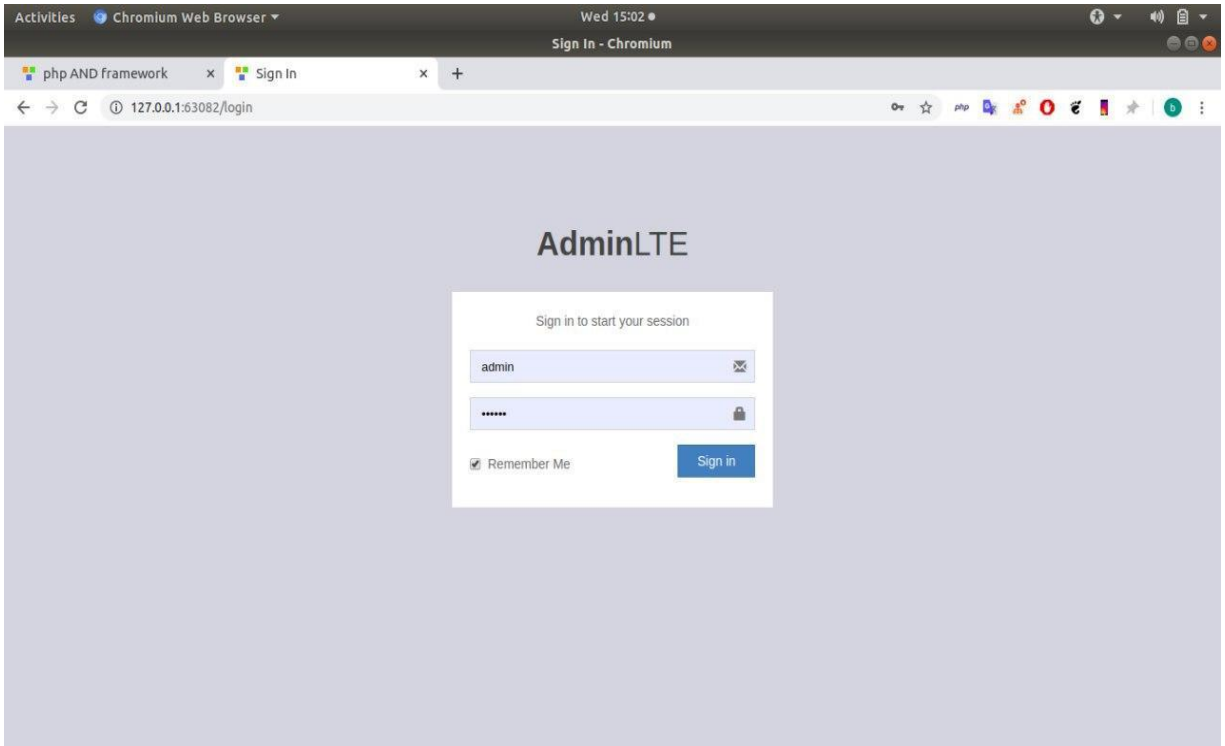
Yii is extremely extensible. You can customize or replace nearly every piece of the core's code. You can also take advantage of Yii's solid extension architecture to use or develop redistributable extensions. High performance is always a primary goal of Yii. Yii is not a one-man show, it is backed up by a strong core developer team, as well as a large community of professionals constantly contributing to Yii's development. The Yii developer team keeps a close eye o.n the latest Web development trends and on the best practices and features found in other frameworks and projects. The most relevant best practices and features found elsewhere are regularly incorporated into the core framework and exposed via simple and elegant interfaces Yii currently has two major versions available: 1.1 and 2.0. Version 1.1 is the old generation and is now in maintenance mode. Version 2.0 is a complete rewrite

of Yii, adopting the latest technologies and protocols, including Composer, PSR, namespaces, traits, and so forth. Version 2.0 represents the current generation of the framework and will receive the main development efforts over the next few years. This guide is mainly about version 2.0. Yii 2.0 requires PHP 5.4.0 or above and runs best with the latest version of PHP 7. You can find more detailed requirements for individual features by running the requirement checker included in every Yii release. Using Yii requires basic knowledge of object-oriented programming (OOP), as Yii is a pure OOP-based framework. Yii 2.0 also makes use of the latest features of PHP, such as namespaces and traits. Understanding these concepts will help you more easily pick up Yii 2.0. You can install Yii in two ways, using the Composer package manager or by downloading an archive file. The former is the preferred way, as it allows you to install new extensions or update Yii by simply running a single command. Standard installations of Yii result in both the framework and a project template being downloaded and installed. A project template is a working Yii project implementing some basic features, such as login, contact form, etc. Its code is organized in a recommended way. Therefore, it can serve as a good starting point for your projects. In this and the next few sections, we will describe how to install Yii with the so-called Basic Project Template and how to implement new features on top of this template. Yii also provides another template called the Advanced Project Template which is better used in a team development environment to develop applications with multiple tiers. Yii 2.0 requires PHP 5.4.0 or above and runs best with the latest version of PHP 7. You can find more detailed requirements for individual features by running the requirement checker included in every Yii release. Using Yii requires basic knowledge of object-oriented programming (OOP), as Yii is a pure OOP-based framework. Yii 2.0 also makes use of the latest features of PHP, such as namespaces and traits. Understanding these concepts will help you more easily pick up Yii 2.0. You can install Yii in two ways, using the Composer package manager or by downloading an archive file. The former is the preferred way, as it allows you to install new extensions or update Yii by simply running a single command. Standard installations of Yii result in both the framework and a project

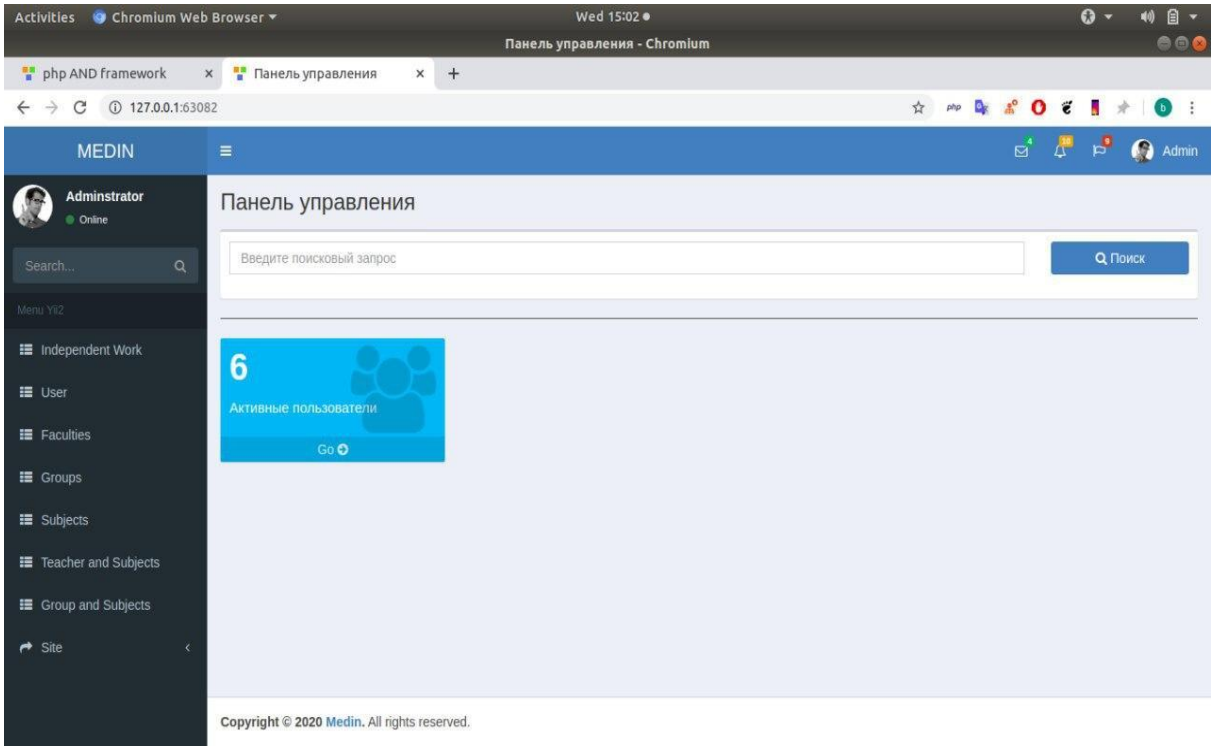
template being downloaded and installed. A project template is a working Yii project implementing some basic features, such as login, contact form, etc. Its code is organized in a recommended way. Therefore, it can serve as a good starting point for your projects. In this and the next few sections, we will describe how to install Yii with the so-called Basic Project Template and how to implement new features on top of this template. Yii also provides another template called the Advanced Project Template which is better used in a team development environment to develo

2.3.Practical part Tasks system

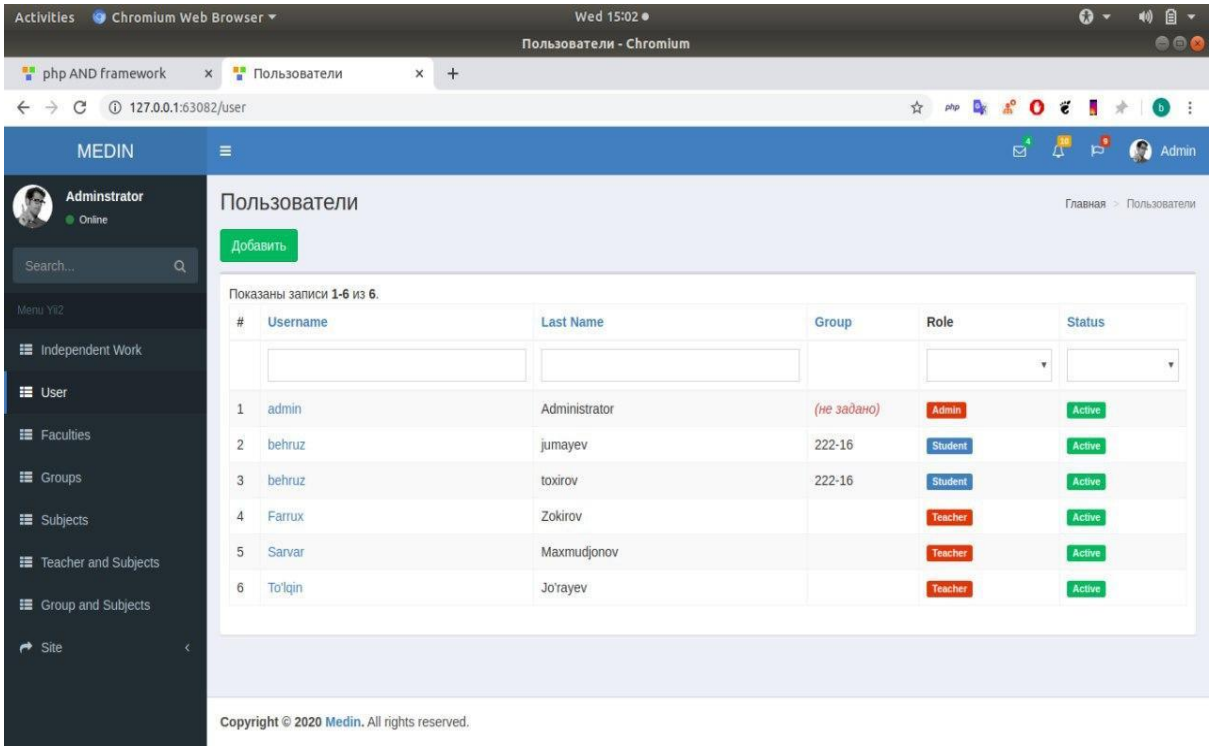
It is no secret Admin need to all system or website. That is why admin need to my task system. Admin have all roles



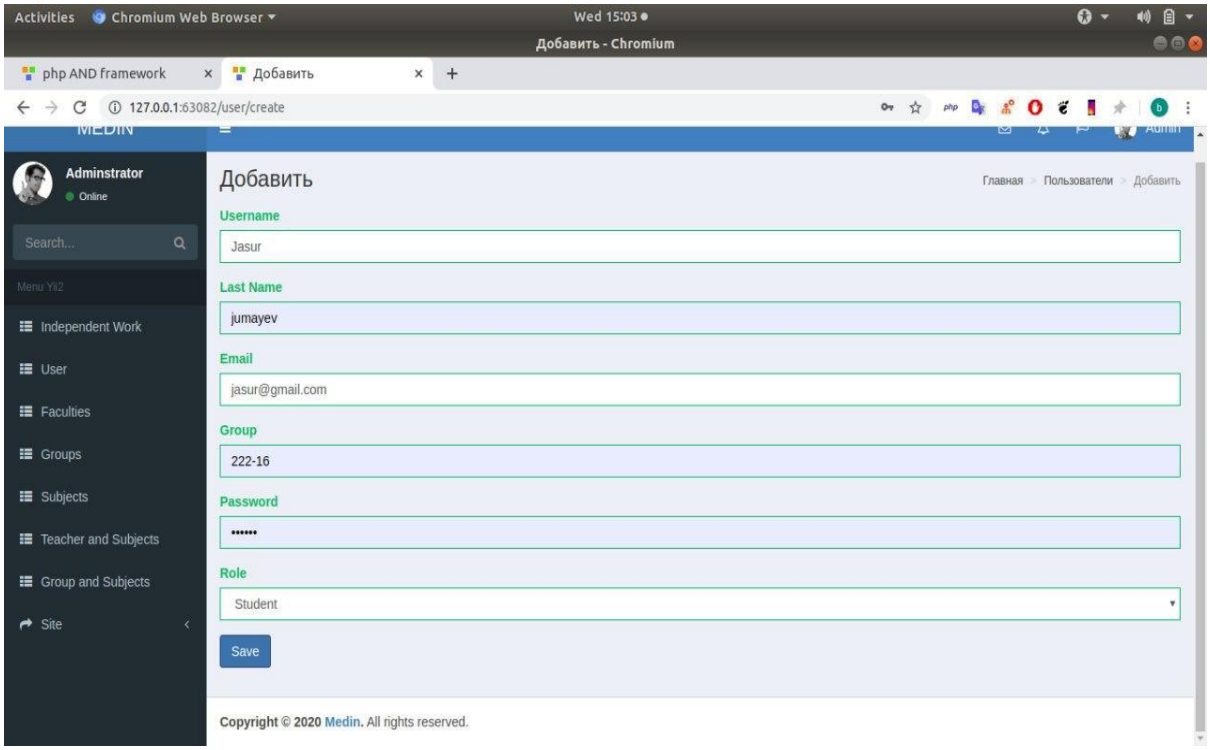
2.1. picture for enter administrator



2.2 Picture Admin panel

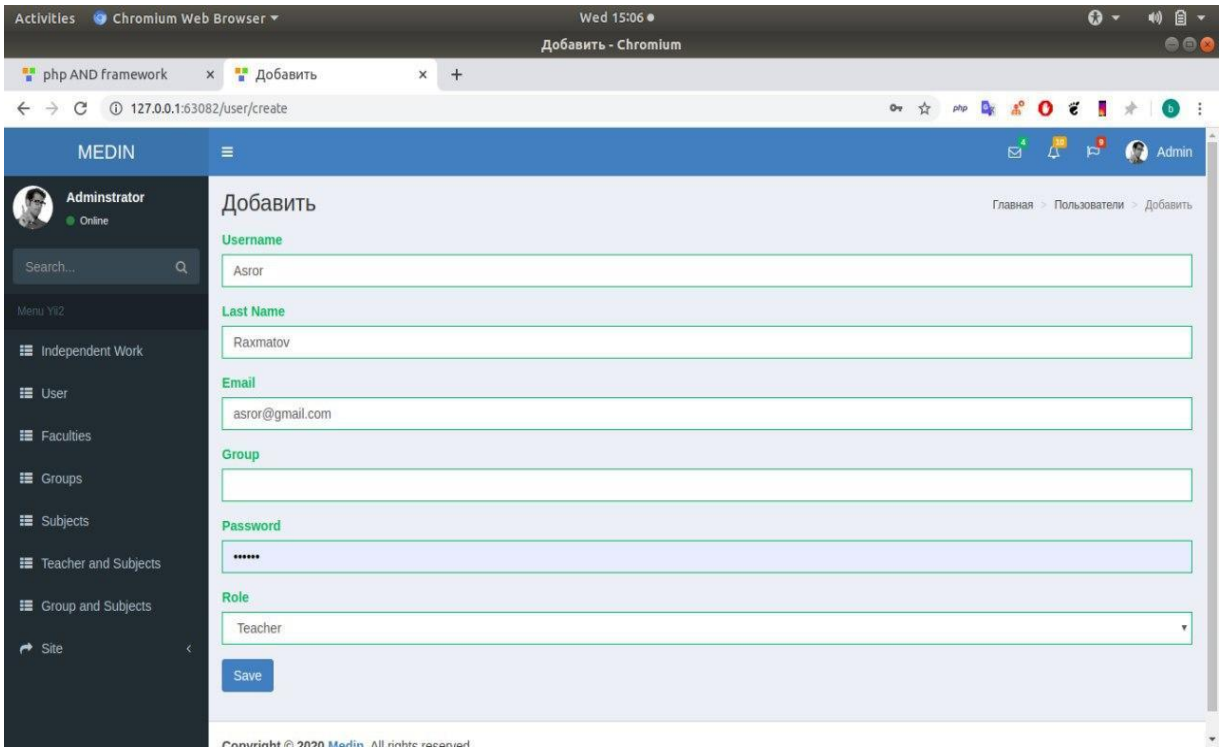


2.3 Picture Users Page

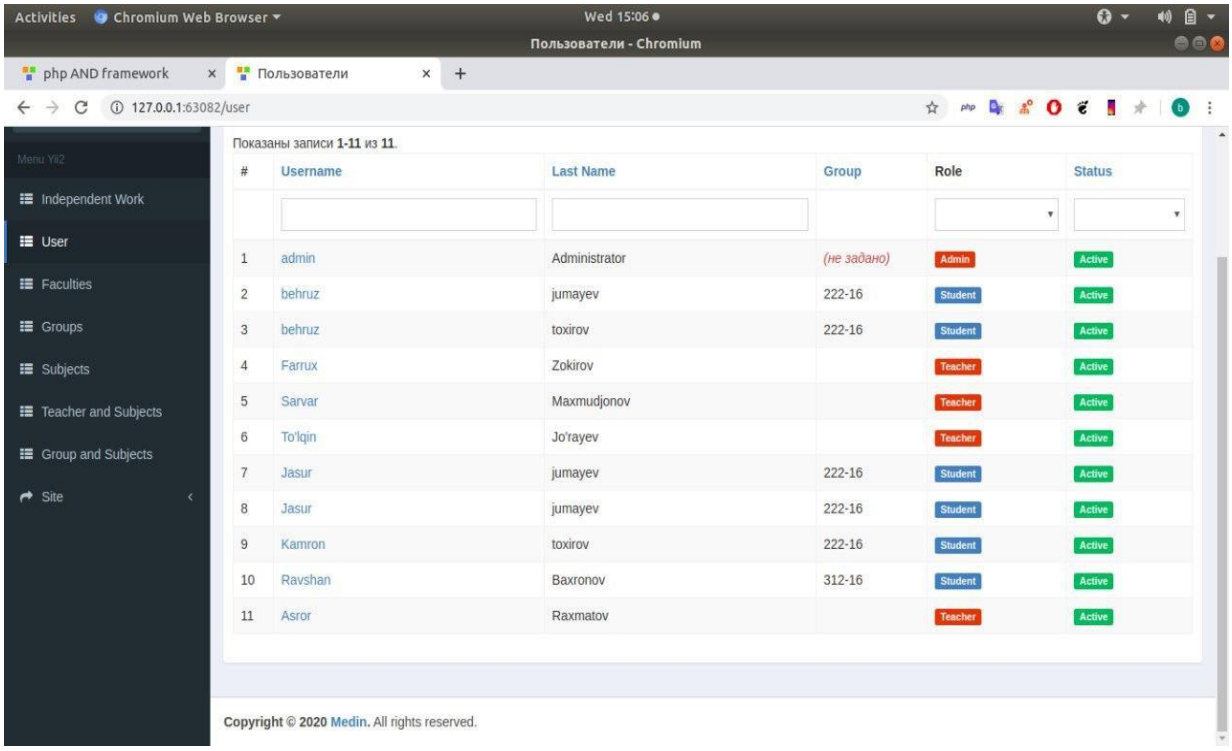


2.4 Picture this page for added Student

On this page, the admin adds students and teachers. and can change their data.

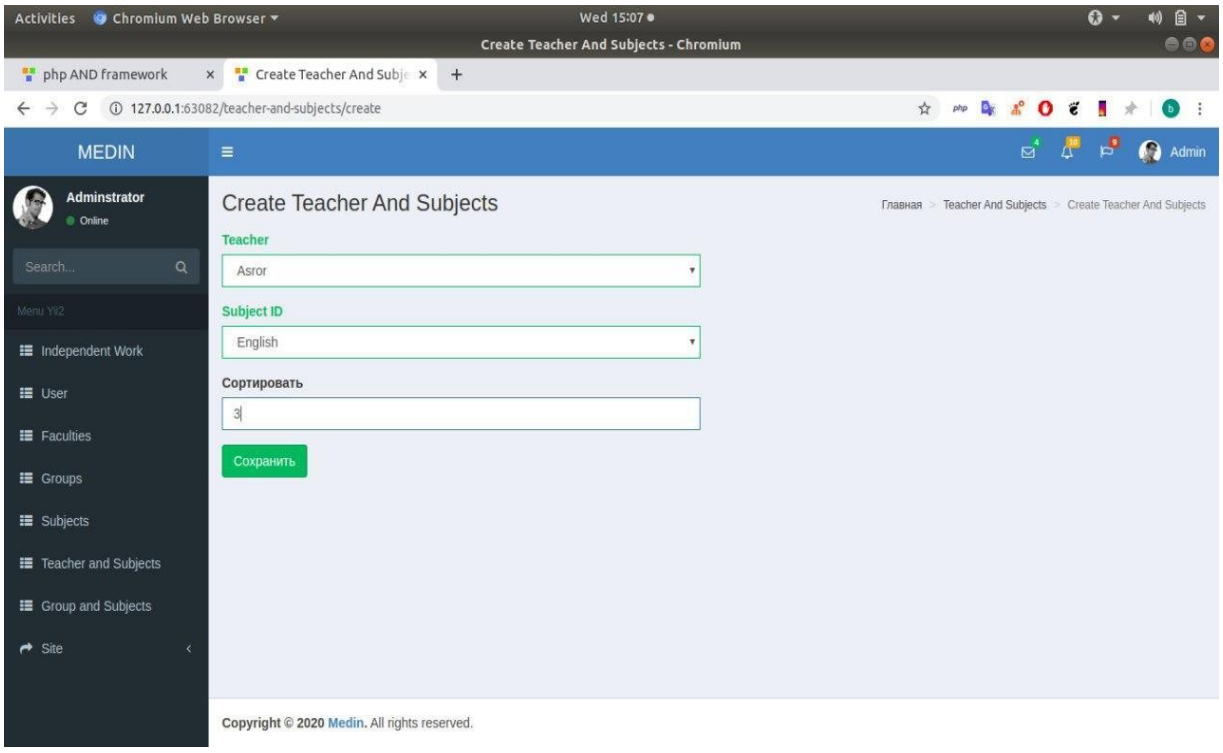


2.5 Picture this page for added Teacher



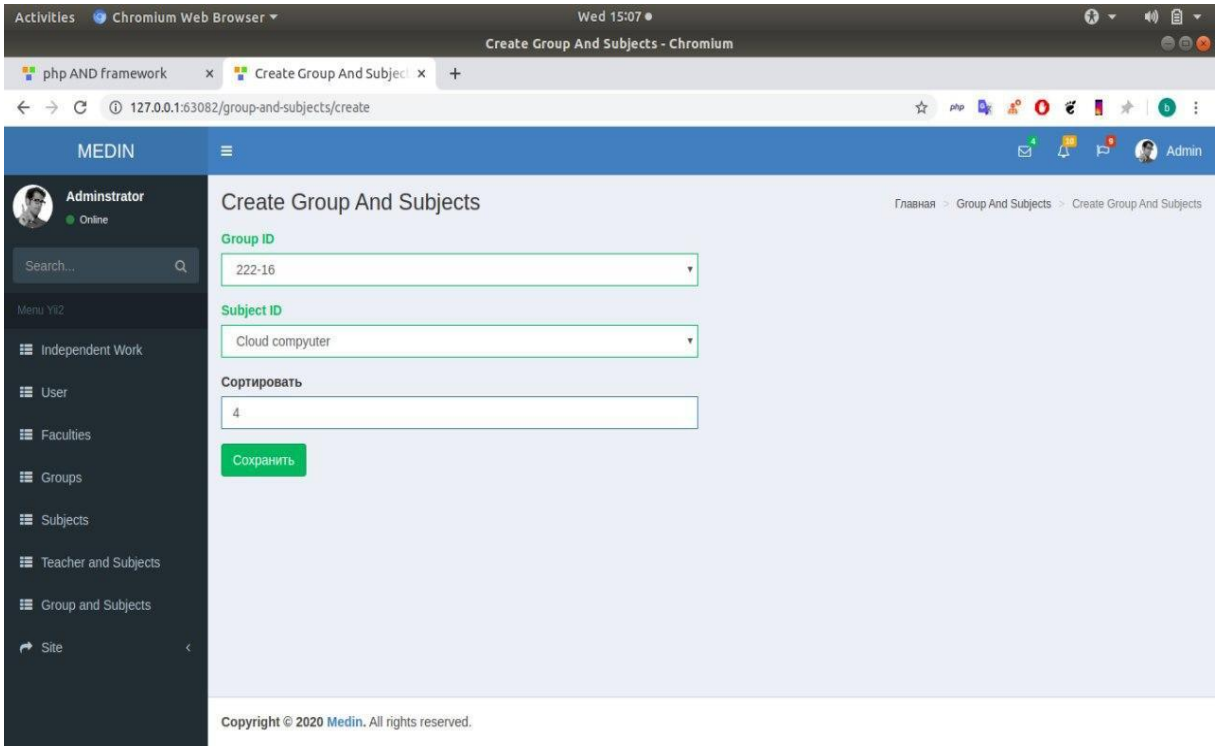
2.6 Picture Users page

This page contains user information. The admin follows the users from this page.



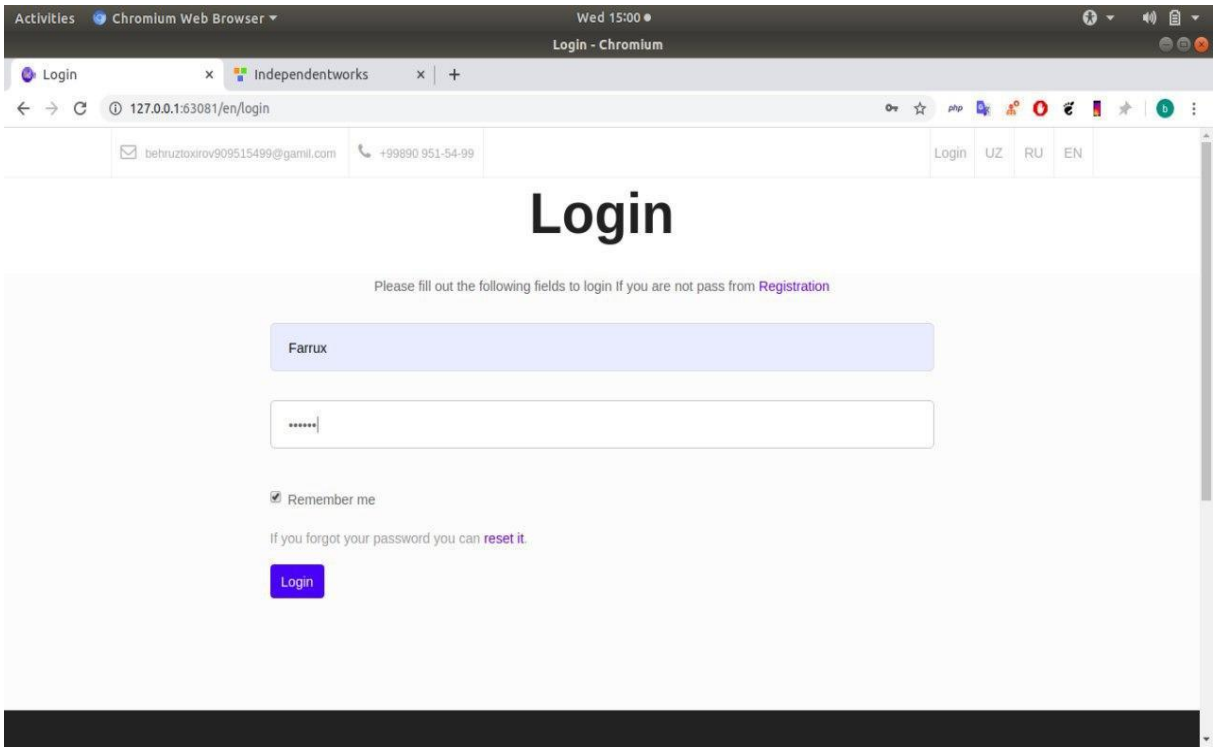
2.7 Picture this page for added subject to teacher

On this page, subjects are attached to teachers. Subjects related to the teacher are attachedandsupervised.



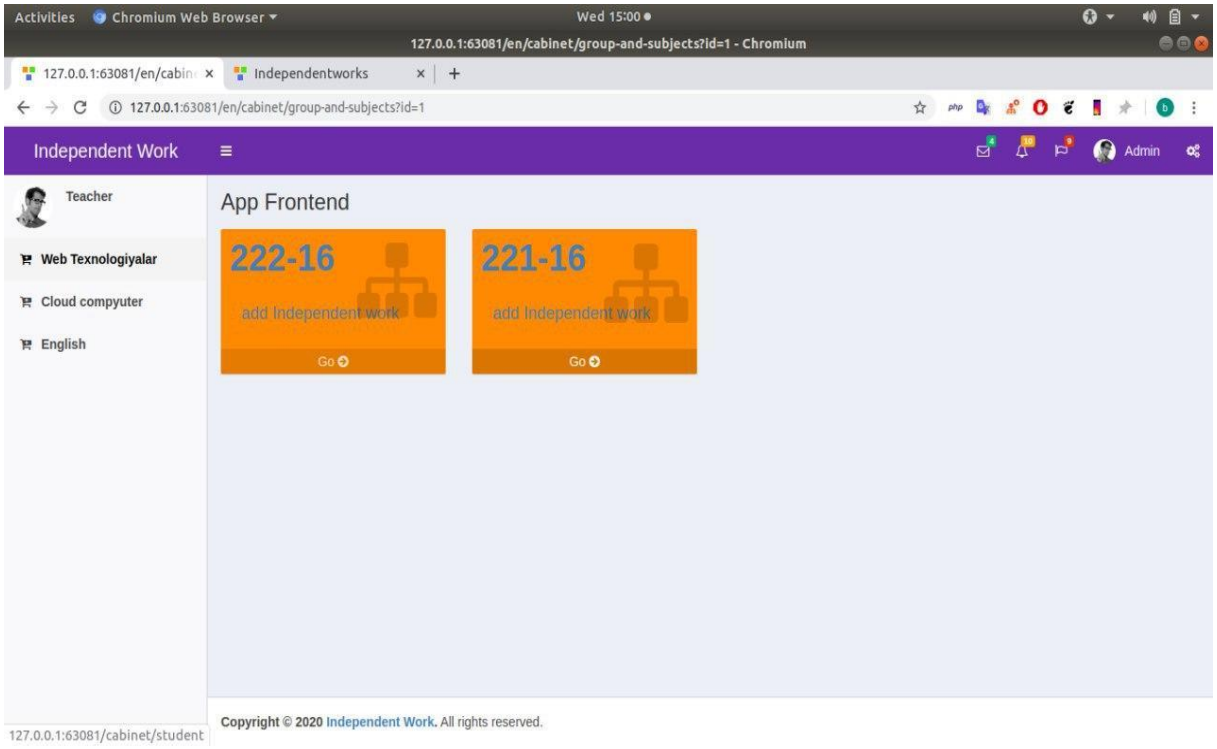
2.8 Picture this page for added subject to group

On this page, the administrator outlines the subjects for the groups. Learns which science to transfer to which group.



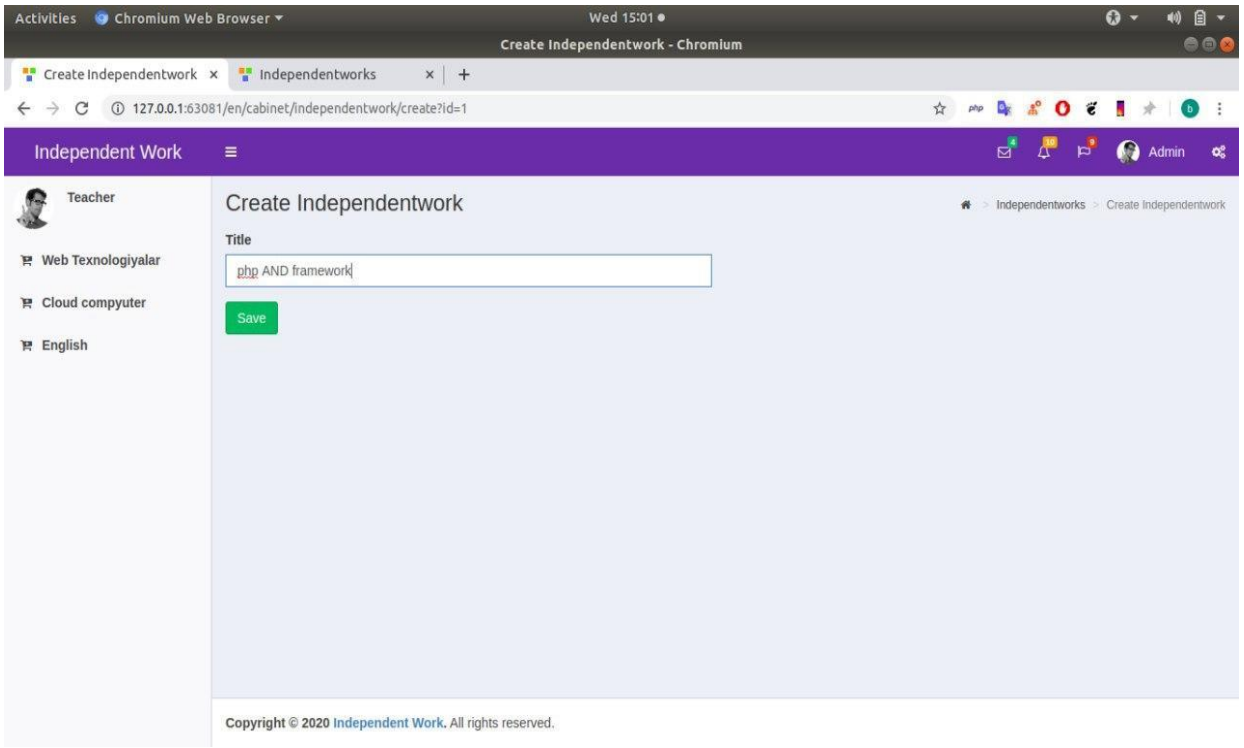
2.9 Picture this login page enter to Tasks system

On this page, users get their rights to login and use

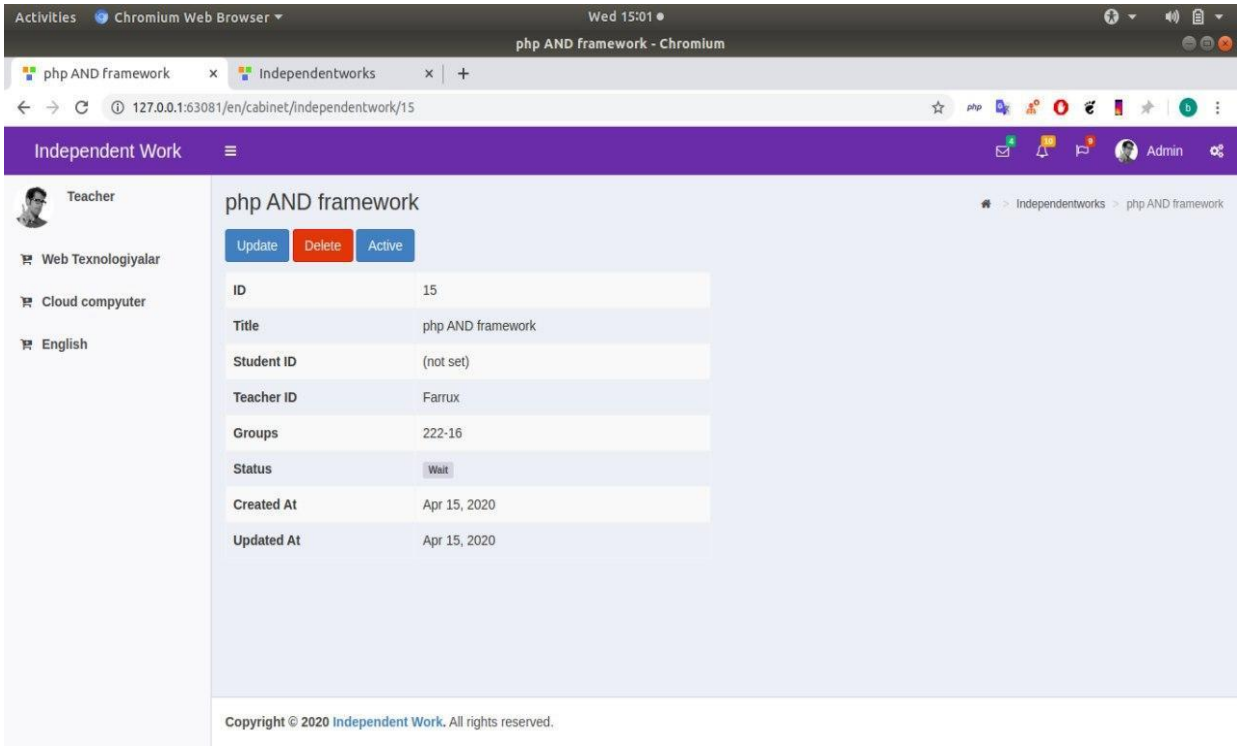


3.0 This page Cabinet for teacher

This page lists relevant subjects and groups for teachers, and assigns work-independent work to their respective groups.

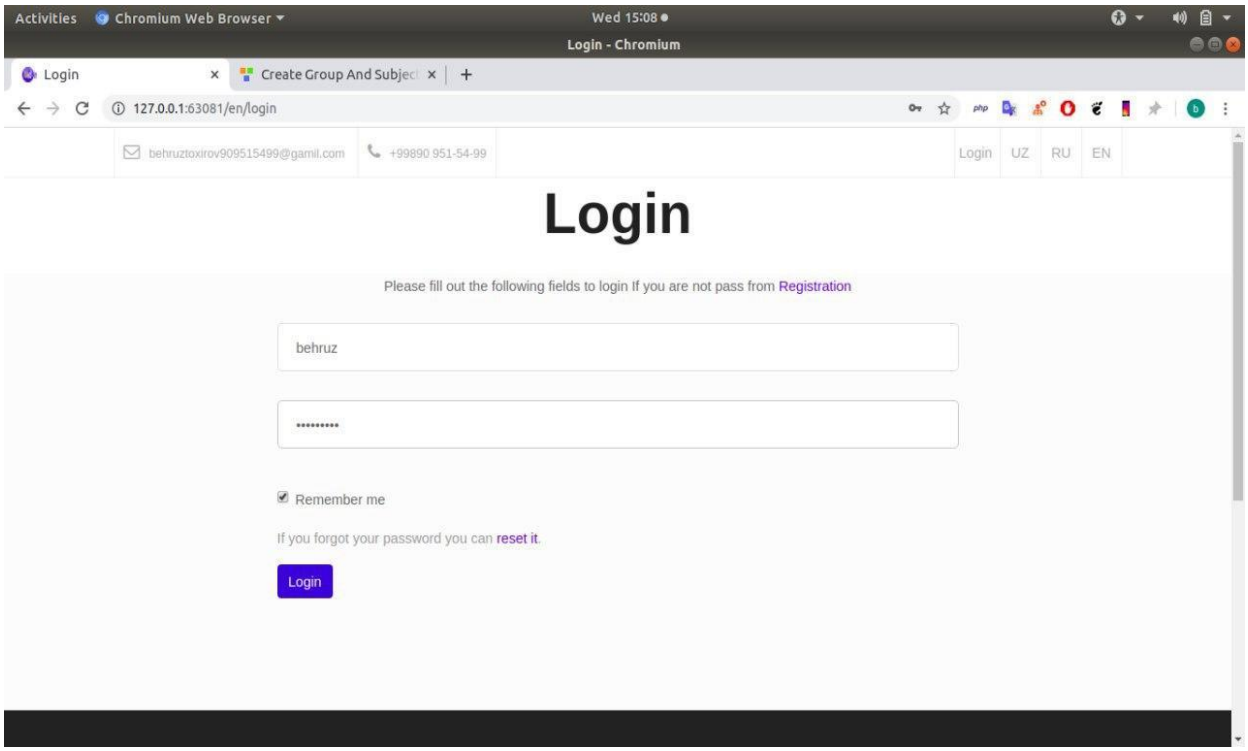


3.1 This page for added independent

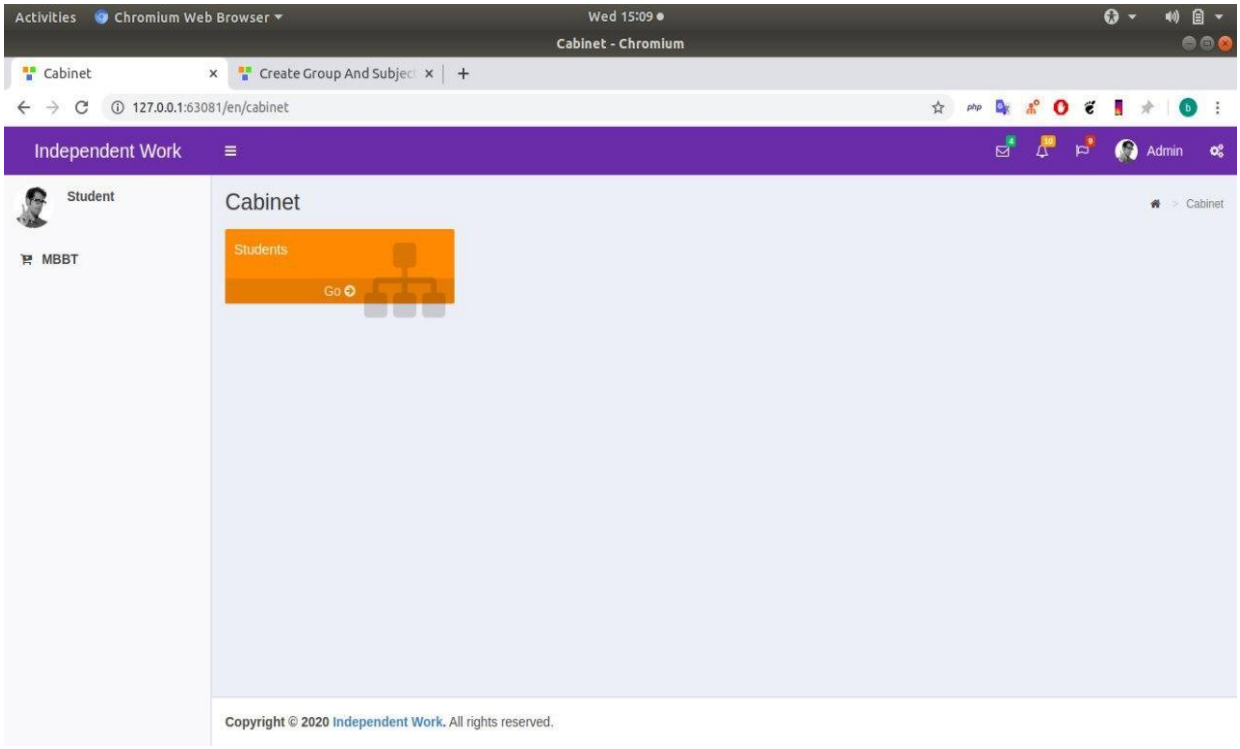


3.2 This page independent page

You can be sure that the independent work entered on this page is loaded.

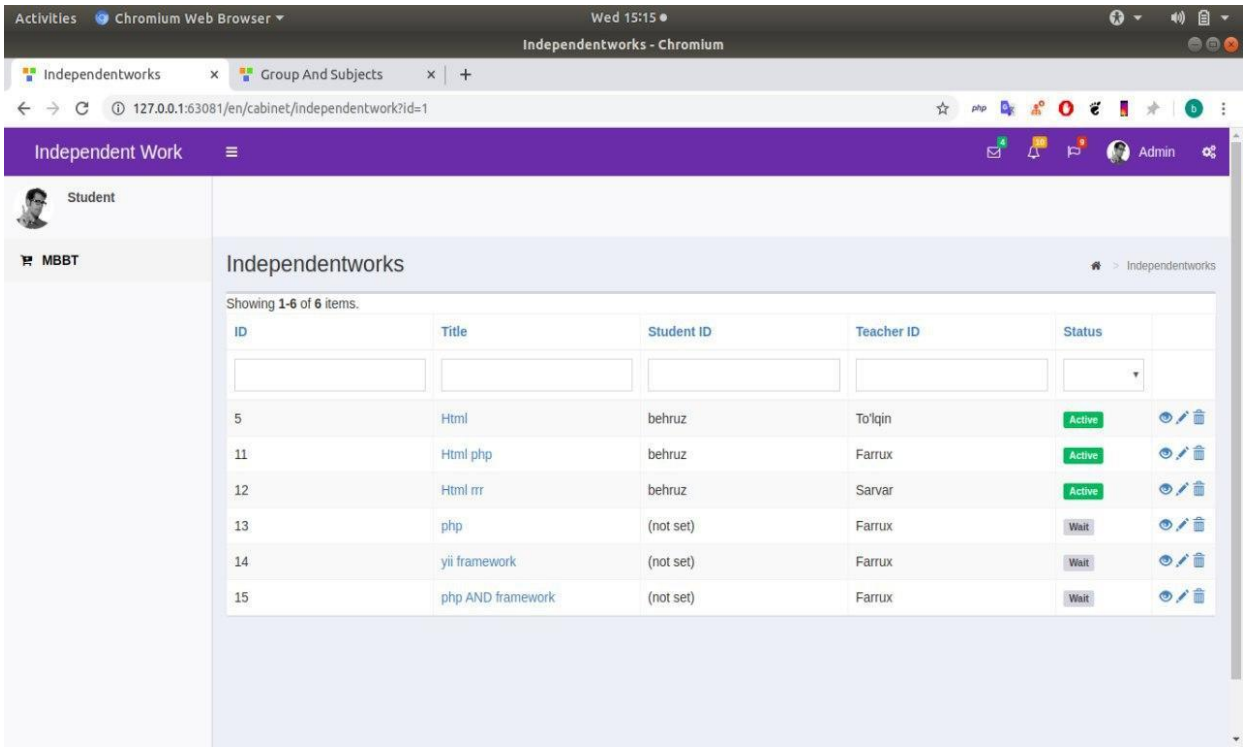


3.3 Picture This page enter to Tasks System for Students

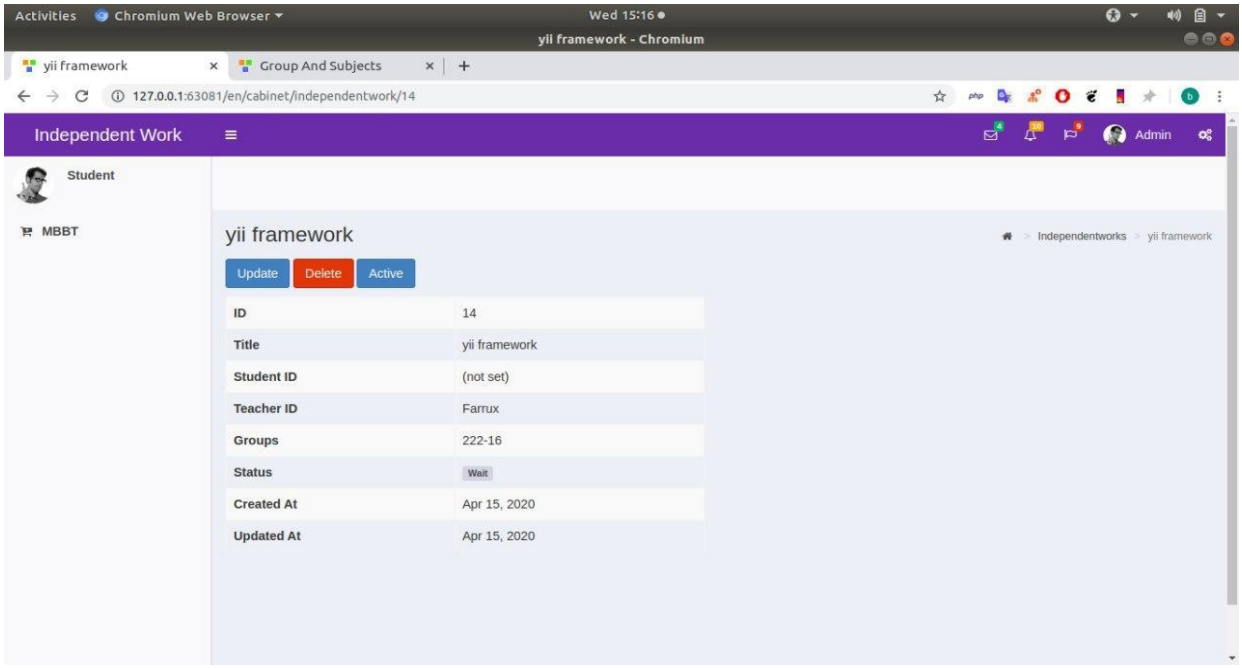


3.4 Picture this Cabinet page for students

On this page, students will find relevant subjects and will be able to enter their own subjects and work independently. It's almost like a teacher's office.

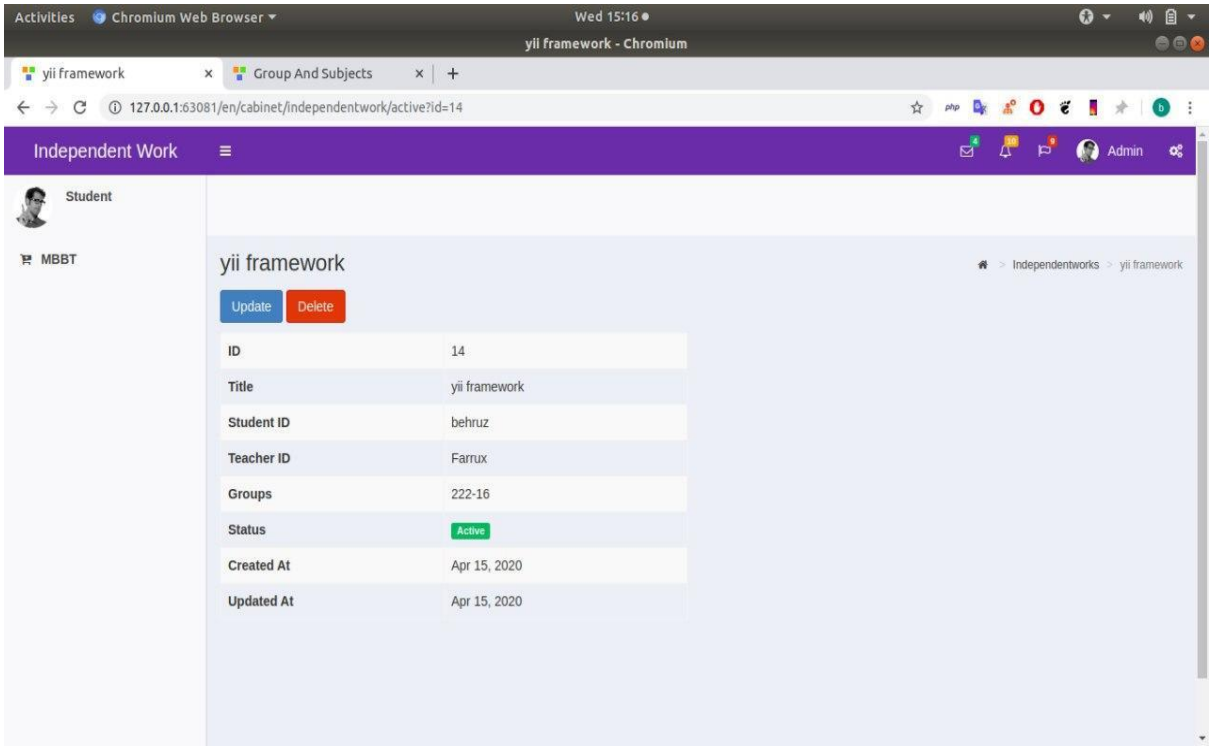


3.5 Picture this page independent works page



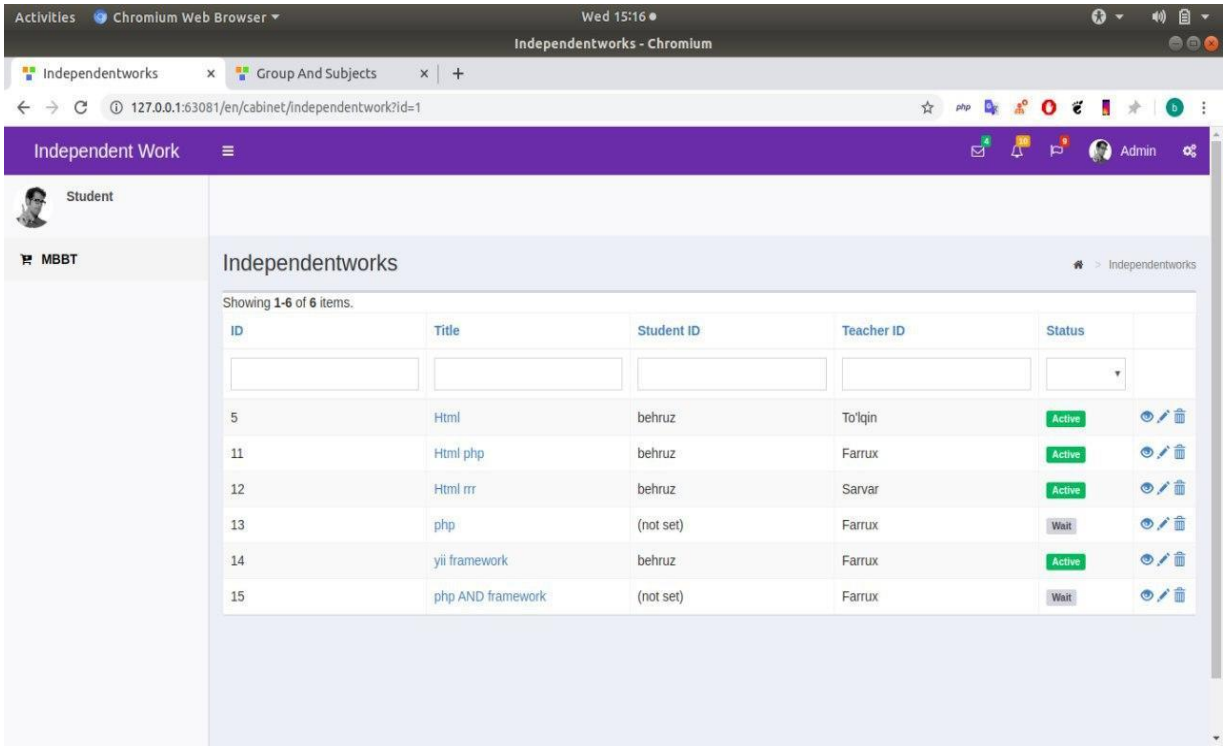
3.6 Picture this page select independent work for student

On this page, the student will have to activate the independent work of their choice, which means that it will be activated so that others will not have to choose again.



3.6 Picture this page selected independent work\

Ko’rib turganingizdek mustaqil ish talabaga biriktirildi



3.6 Picture this page independent works page

Bu sahifada esa talabalar tanlagan mustaqil ishlari aktiv holat tanlanmaganlari esa kutish tizimida turipti

CONCLUSION

While working with graduation work, I used programs framework of PHP. I learned that interactivity is the interaction of a person and an object. I learned about tasks systems. Having studied the task system of different countries, I created an interactive technology task system where people could find out what old technologies were like and how they worked. In my graduation work, not only old but also new technologies are presented, so that people know how much has changed. The following tasks were set and completed:

1. I learned useful systems for higher education.
2. By doing this system, I have further strengthened my experience in programming.
3. I strengthened my knowledge by using web technologies in the system.

REFERENCES

1. President of the Republic of Uzbekistan Command No. PP-3245 dated 29.08.2017 y. 2. Association of Art Museum Directors. (2015). Next Practices in Digital and Technology. Retrieved December 4, 2016.

3. Bernardo, Á. (2016). Museums that are Accessible Thanks to Technology. Retrieved December 19, 2016.

4. Space planning and room design on a computer. Author: Lebedev A. Year of publication: 2011

5. Yii framework Author: Pekarev Leonid, Year of publication: 2011

6. Lumion yii2 framework. Type of publication: Separate edition. Year of release 2014. Ciro Cardoso Author.

7. framework of PHP. Type of publication: Separate edition. Year of release 2014. Ciro Cardoso Author.

8. http://www.all-ebooks.com/

9. http://www.nytimes.com/

10. https://aamd.org/

11. https://www.google.com/

12. https://www.youtube.com/ 80

13. How to Use Graphic Design to Sell Things, Explain Things, Make Things Look Better, Make People Laugh, Make People Cry, and (Every Once in a While) Change the World – Michael Bierut page-350

14. Ready to Print : Handbook for Media Designers – Kristina Nickel page-258

15. Manuals 2: Design and Identity Guidelines page-350

16. The Design Annual 2015 Edition page-300

17. Drawing Type: An Introduction to Illustrating Letterforms – Alex Fowkes page - 420

18. http://gov.uz/ (Ўзбекистон Республикасининг Ҳукумат портали).

19. http://lex.uz/ (Ўзбекистон Республикаси Қонун ҳужжатлари маълумотлари миллий базаси).

```
<?php
use backend\models\GroupAndSubjects;
$user_id = Yii::$app->user->id;
$usergroup = \independentwork\entities\User\User::find()->where(['id' => $user_id])-
>one();
if ((Yii::$app->user->can('teacher'))) {
   $teacher_subjects = backend\models\TeacherAndSubjects::find()->where(['user_id'
=> $user_id])->all();
   $subjects = backend\models\Subjects::find()->all();
   $teacherItems = [];
   foreach ($teacher_subjects as $ts):?>
      <?php foreach ($subjects as $subject): ?>
         <?php if ($subject['id'] == $ts->subject_id): ?>
            <?php $teacherItems[] = ['label' => $subject->title, 'icon' => 'cart-plus', 'url'
=> ['cabinet/group-and-subjects/index', 'id' => $subject->id]]; ?>
         <?php endif; ?>
      <?php endforeach; ?>
   <?php endforeach; }
if ((Yii::$app->user->can('student'))) {
   $query = GroupAndSubjects::find()->JoinWith(['group g'], true, 'INNER JOIN')-
>where(['g.title' => $usergroup->group])->all();
   foreach ($query as $q):endforeach
   $subjects = backend\models\Subjects::find()->where(['id' => $q->subject_id])->all();
   $studentItems = [];    foreach ($query as $gs):?>
      <?php foreach ($subjects as $subject): ?>
         <?php if ($subject['id'] == $gs->subject_id): ?>
            <a href="<?= \yii\helpers\Url::to(['cabinet/independentwork/index', 'id' =>
$subject->id]) ?>">
              <h2><?= $subject->title ?></h2></a>
            <?php $studentItems[] = ['label' => $subject->title, 'icon' => 'cart-plus', 'url'
=> ['/cabinet/independentwork/index']]; ?>
         <?php endif; ?>
      <?php endforeach; ?>
   <?php endforeach; }
$studentItems = (Yii::$app->user->can('student')) ? $studentItems : [];
$teacherItems = (Yii::$app->user->can('teacher')) ? $teacherItems : []; ?>
<aside class="main-sidebar">
   <section class="sidebar">
      <div class="user-panel">
         <div class="pull-left image">
            <img src="<?= $directoryAsset ?>/img/user2-160x160.jpg" class="img-
circle" alt="User Image"/>
         </div>
```

```
            <div class="pull-left info">
                <p><?php if (Yii::$app->user->can('student')) {
                        echo 'Student';
                    } else {
                        echo 'Teacher';
                    } ?>
                </p>
            </div>
        </div>
        <?= dmstr\widgets\Menu::widget(             [
                'options' => ['class' => 'sidebar-menu tree', 'data-widget' => 'tree'],
                'items' => array_merge($teacherItems, $studentItems),
            ]       ) ?    </section></aside><?php
/* @var $this yii\web\View */
use backend\forms\UserSearch;
use yii\helpers\Url;
$this->title = 'Cabinet';
$this->params['breadcrumbs'][] = $this->title;
$searchModel = new UserSearch();;?>
<?php if (Yii::$app->user->can('student')): ?>
    <div class="row">
        <div class="cabinet-index">
            <div class="col-lg-3 col-xs-6">
                <div class="small-box bg-orange">
                    <div class="inner">
                        <p><?= Yii::t('app', 'Students') ?></p>
                    </div>
                    <div class="icon"><i class="fa fa-sitemap"></i></div>
                    <a href="<?= Url::to('/cabinet/student') ?>" class="small-box-footer"><?=
Yii::t('app', 'Go') ?> <i   class="fa fa-arrow-circle-right"></i></a>
                </div>          </div>       </div>    </div>
<?php endif; ?>
<?php if (Yii::$app->user->can('teacher')): ?>
    <div class="row">
        <div class="cabinet-index">
            <div class="col-lg-3 col-xs-6">
                <div class="small-box bg-orange">
                    <div class="inner">
                        <p><?= Yii::t('app', 'Teachers') ?></p><br>
                        <p></p>
                    </div>
                    <div class="icon"><i class="fa fa-sitemap"></i></div>
                    <a href="<?= Url::to('/cabinet/teacher') ?>" class="small-box-footer"><?=
Yii::t('app', 'Go') ?> <I class="fa fa-arrow-circle-right"></i></a>              </div>
            </div>
```

```
        </div>    </div><?php endif; ?>
?php
namespace frontend\controllers\cabinet;
use backend\models\Independentwork;
use frontend\forms\IndependentworkSearch;
use Yii;
use yii\filters\VerbFilter;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
/
 * IndependentworkController implements the CRUD actions for Independentwork
model.
 */
class IndependentworkController extends Controller
{
    const STATUS_WAIT = 0;
    const STATUS_ACTIVE = 10;
    public $layout = 'cabinet';
    /
     * {@inheritdoc}
     */
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ];
    }
    /
     * Lists all Independentwork models.
     * @return mixed
     */
    public function actionIndex($id)
    {
        $searchModel = new IndependentworkSearch();
        $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
        $dataProvider->query->where(['group_id' => $id]);
        return $this->render('index', [
            'searchModel' => $searchModel,
            'dataProvider' => $dataProvider,
        ]);
```

```
    }
    /
     * Displays a single Independentwork model.
     * @param integer $id
     * @return mixed
     * @throws NotFoundHttpException if the model cannot be found
     */
    public function actionView($id)
    {
        return $this->render('view', [
            'model' => $this->findModel($id),
        ]);
    }
    /
     * Finds the Independentwork model based on its primary key value.
     * If the model is not found, a 404 HTTP exception will be thrown.
     * @param integer $id
     * @return Independentwork the loaded model
     * @throws NotFoundHttpException if the model cannot be found
     */
    protected function findModel($id)
    {
        if (($model = Independentwork::findOne($id)) !== null) {
            return $model;
        }

        throw new NotFoundHttpException(Yii::t('app', 'The requested page does not
exist.'));
    }
    /
     * Creates a new Independentwork model.
     * If creation is successful, the browser will be redirected to the 'view' page.
     * @return mixed
     */
    public function actionCreate($id)
    {
        $model = new Independentwork();

        if ($model->load(Yii::$app->request->post())) {
            $model->group_id = $id;
            $model->teacher_id = Yii::$app->user->id;
            $model->save();
            return $this->redirect(['view', 'id' => $model->id]);
        }
```

```
        return $this->render('create', [
            'model' => $model,
        ]);
    }
    /
     * Updates an existing Independentwork model.
     * If update is successful, the browser will be redirected to the 'view' page.
     * @param integer $id
     * @return mixed
     * @throws NotFoundHttpException if the model cannot be found
     */
    public function actionUpdate($id)
    {
        $model = $this->findModel($id);
        if ($model->load(Yii::$app->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }

        return $this->render('update', [
            'model' => $model,
        ]);
    }
    /
     * Deletes an existing Independentwork model.
     * If deletion is successful, the browser will be redirected to the 'index' page.
     * @param integer $id
     * @return mixed
     * @throws NotFoundHttpException if the model cannot be found
     */
    public function actionDelete($id)
    {
        $this->findModel($id)->delete();

        return $this->redirect(['index']);
    }
    public function actionActive($id)
    {
        $independentwork = Independentwork::find()->where(['id' => $id])->one();
        $independentwork->status = self::STATUS_ACTIVE;
        $independentwork->student_id = Yii::$app->user->getId();
        $independentwork->save();
        return $this->render('view', [
            'id' => $id,
            'model' => $independentwork,
        ]);
```

```
    }
}
<?php
namespace frontend\controllers\cabinet;
use backed\models\GroupAndSubjects;
use backend\models\Groups;
use yii\filters\VerbFilter;
use yii\web\Controller;
/
 * GroupAndSubjectsController implements the CRUD actions for GroupAndSubjects
model.
 */
class GroupAndSubjectsController extends Controller
{
    public $layout = 'cabinet';
    /
     * {@inheritdoc}
     */
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ];
    }
    public function actionIndex($id)
    {
        $group_subject = GroupAndSubjects::find()->where(['subject_id' => $id])->all();
        $groups = Groups::find()->all();

        return $this->render('index', [
            'group_subjects' => $group_subject,
            'groups' => $groups,
        ]);
    }
    public function actionView()
    {
        return $this->render('view', [

        ]);
    }
```

```
}
<?php
namespace frontend\controllers\cabinet;
use bckend\models\TeacherAndSubjects;
use Yii;
use backend\models\Subjects;
use yii\web\Controller;
use yii\filters\VerbFilter;
/
 * SubjectsController implements the CRUD actions for Subjects model.
 */
class SubjectsController extends Controller
{
    public $layout = 'cabinet';
    /
     * {@inheritdoc}
     */
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ];
    }
    public function actionIndex()
    {
        $user_id = Yii::$app->user->id;

        $teacher_subjects = TeacherAndSubjects::find()->where(['user_id' => $user_id])-
>all();
        $subjects = Subjects::find()->all();
        return $this->render('index', [

        ]);
    }
    public function actionView($id)
    {
        return $this->render('view', [
```

```
        ]);
    }
}
<?php
use abdualiym\cms\entities\Menu;
use abdualiym\cms\helpers\Language;
use frontend\widgets\LanguagesWidget;
/**
 *
 * @var $menu Menu
 */
$url = Yii::$app->language;
?>
<div class="menu-area rs-defult-header menu-sticky">
    <div class="container">
        <div class="row align-items-center">
            <div class="col-lg-2">
                <div class="logo-area">
                </div>
            </div>
            <div class="col-lg-10">
                <div class="main-menu">
                    <a class="rs-menu-toggle"><i class="fa fa-bars"></i></a>
                    <nav class="rs-menu">
                        <ul class="nav-menu">
                            <?php foreach ($menu as $key => $value) {
                                if ($value['type'] != Menu::TYPE_EMPTY) {
                                    echo '<li class="rs-mega-menu current-menu-item menu-item-
has-children"><a class="active" href="' . $value['link'] . '">' .
Language::getAttribute($value, 'title', $url) . '</a></li>';
                                } else {
                                    if (isset($value['childs']) && $value['childs']) {
                                        echo '<li class="menu-item-has-children">' . '<a href="' .
$value['link'] . '">' . Language::getAttribute($value, 'title', $url) . ' <i class="fa fa-angle-
down"></i></a>';
                                        echo '<ul class="sub-menu">';
                                        foreach ($value['childs'] as $key => $childValue) {
                                            if (isset($childValue['childs']) && $childValue['childs']) {
                                                //echo '<li class="menu-item-has-children">' . '<a href="'
. $childValue['link'] . '">' . Language::getAttribute($childValue, 'title', $url) . ' <i
class="fa fa-angle-down"></i></a>';

                                                echo '<li class="menu-item-has-children">' . '<a href="' .
$childValue['link'] . '">' . Language::getAttribute($childValue, 'title', $url) . ' <i
class="fa fa-angle-down"></i></a>';
```

```
echo '<ul class="sub-menu">';
                                    foreach ($childValue['childs'] as $key => $children) {
                                        if (isset($children['childs']) && $children['childs']) {
                                            echo '<li class="menu-item-has-children">' . '<a
href="' . $childValue['link'] . '">' . Language::getAttribute($childValue, 'title', $url) . ' <i
class="fa fa-angle-down"></i></a>';
                                            echo '<ul class="sub-menu">';
                                            foreach ($children['childs'] as $key => $children3) {
                                                echo '<li><a href="' . $children3['link'] . '">' .
Language::getAttribute($children3, 'title', $url) . '</a></li>';
                                            }
                                            echo '</ul>';
                                            echo '</li>';
                                        } else {
                                            echo '<li><a href="' . $children['link'] . '">' .
Language::getAttribute($children, 'title', $url) . '</a></li>';
                                        }
                                    }
                                    echo '</ul>';
                                    echo '</li>';
                                } else {
                                    echo '<li><a href="' . $childValue['link'] . '">' .
Language::getAttribute($childValue, 'title', $url) . '</a></li>';
                                }
                            }
                            echo '</ul>';
                            echo '</li>';
                        } else {
                            echo '<li><a href="' . $value['link'] . '">' .
Language::getAttribute($value, 'title', $url) . '</a></li>';
                        }
                    }
                }
                ?>
                <li class="menu-item-has-children hidden-md hidden-lg"><a
href="#"><?= Yii::t('app', 'Login') ?><i class="fa fa-angle-down"></i></a>
                    <ul class="sub-menu">
                        <li><a href=""><?= Yii::t('app', 'Login') ?></a></li>
                    </ul>
                </li>
            </ul>
        </nav>
    </div>
</div>
</div>
```

```
    </div>
</div>
<?php
/* @var $this View */
/* @var $content string */
use frontend\assets\AppAsset;
use frontend\widgets\CanvasWidget;
use frontend\widgets\FooterWidget;
use frontend\widgets\MenuWidget;
use frontend\widgets\PartnerWidget;
use frontend\widgets\ToolbarWidget;
use yii\helpers\Html;
use yii\web\View;
AppAsset::register($this);
?>
<?php $this->beginPage() ?>
<!DOCTYPE html>
<html lang="<?= Yii::$app->language ?>">
<head>
    <meta charset="<?= Yii::$app->charset ?>">
    <?php $this->registerCsrfMetaTags() ?>
    <title><?= Html::encode($this->title) ?></title>
    <meta name="description" content="">
    <!-- responsive tag -->
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- favicon -->
    <link rel="apple-touch-icon" href="/images/logoo.png">
    <link rel="shortcut icon" type="image/x-icon" href="/images/logoo.png">
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
    <?php $this->head() ?>
</head>
<body class="defult-home">
<?php $this->beginBody() ?>
<!--Header Start-->
<header id="rs-header" class="rs-header">
    <!-- Toolbar Start -->
    <?= ToolbarWidget::widget() ?>
    <!-- Toolbar End -->
    <!-- Header Menu Start -->
    <?= MenuWidget::widget() ?>
    <!-- Header Menu End -->
```

```
    <!-- Canvas Menu start -->
    <?= CanvasWidget::widget() ?>
    <!-- Canvas Menu end -->
</header>
<!--Header End-->
<!-- Main content Start -->
<div class="main-content">
    <?= $content ?>
</div>
<!-- Main content End -->
<!-- Footer Start -->
<?= FooterWidget::widget() ?>
<!-- Footer End -->
<!-- start scrollUp  -->
<div id="scrollUp">
    <i class="fa fa-angle-up"></i>
</div>
<?php $this->endBody() ?>
</body>
</html>
<?php $this->endPage() ?><?ph
namespace frontend\forms;
use backend\models\Groups;
use backend\models\Independentwork;
use Yii;
use yii\base\Model;
use yii\data\ActiveDataProvider;
/
 * IndependentworkSearch represents the model behind the search form of
`backend\models\Independentwork`.
 */
class IndependentworkSearch extends Independentwork
{
    /
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['id', 'student_id', 'teacher_id', 'status', 'created_at', 'updated_at'], 'integer'],
            [['title'], 'safe'],
        ];
    }
```

```
    /
     * {@inheritdoc}
     */
    public function scenarios()
    {
        // bypass scenarios() implementation in the parent class
        return Model::scenarios();
    }
    /
     * Creates data provider instance with search query applied
     *
     * @param array $params
     *
     * @return ActiveDataProvider
     */
    public function search($params)
    {
        $query = Independentwork::find();
        if (Yii::$app->user->can('student')) {
            $user_id = Yii::$app->user->id;
            $usergroup = \independentwork\entities\User\User::find()->where(['id' =>
$user_id])->one();

            $group = Groups::find()->where(['title' => $usergroup->group])->one();
            $query = Independentwork::find()->where(['group_id' => $group->id]);
        } else if (Yii::$app->user->can('teacher')) {
            $user_id = Yii::$app->user->id;

            $query = Independentwork::find()->where(['teacher_id' => $user_id]);
        }
        $dataProvider = new ActiveDataProvider([
            'query' => $query,
        ]);

        $this->load($params);

        if (!$this->validate()) {

            return $dataProvider;
        }
        // grid filtering conditions
        $query->andFilterWhere([
            'id' => $this->id,
            'student_id' => $this->student_id,
```

```
            'teacher_id' => $this->teacher_id,
            'status' => $this->status,
            'group_id' => $this->group_id,
            'created_at' => $this->created_at,
            'updated_at' => $this->updated_at,
        ]);
        $query->andFilterWhere(['like', 'title', $this->title]);
        return $dataProvider;
    }
}
```